

RELEASE 1.0

4-91

A MONTHLY REPORT BY ESTHER DYSON

30 APRIL 1991

4-91

30 April 1991

CLIENT-SERVER STANDARDS FOR TEXT: FOUNDATION FOR INNOVATION

How do I access thee? Let me count the ways... Dow Jones News Retrieval has one interface; Lotus Magellan another; CompuServe discussion groups a third; our wp files a fourth; Computer Library's Computer Select and Information Access's Magazine Rack (from the same parent company, Ziff Communications!) yet two more. Then there's IZE and Lotus Notes, Folio Views and cc:Mail, ZyIndex and The WELL.

All this at a time when a single user interface (that is, any of many user interfaces) offers access to a wide variety of structured data sources, and a single data source can be addressed through many user interfaces. The promise of SQL -- heterogeneous access to structured data -- is now being realized, and makes the limitations of text retrieval more apparent. Over the next decade we will need to handle a rapidly increasing volume both of unstructured text and of text structured in clever, nonstandard ways by people and by products such as Notes, Verity's Topic, Folio Views, and tools for building semi-structured e-mail messages, forms and EDI applications.

This issue is about some early efforts to provide SQL-like facilities for text -- but remember that it took a decade for SQL to catch on. Perhaps we can do it faster the second time around, as information proliferates and we demand maps and signposts for all the territory in our electronic frontier.

The goal is that a given text front-end can retrieve data from any back-end, instead of the situation now where we have the confusion of front-ends described above. As with data, you should be able to run a single query against your own files, against structured corporate text bases and against external sources such as Dow Jones, Reuters or Mead's Lexis.

The data world has long had SQL (Structured Query Language), a neutral language (and an official standard) for describing databases and querying data that works across platforms and databases. Detractors point out that SQL is only a subset of a multitude of diverse systems that don't interoperate. It's a description language, not a programming language, and can't do much by itself. But of course that's also its virtue. People have been innovating around SQL for the past decade and will continue to do so well into the 21st century.

It's much harder to develop standards for communication between client and server for text since →

THE TRANSCRIPTS ARE COMING!

INSIDE	
SQL FOR TEXT	1
<i>Serve me some text.</i>	
<i>WAIS has many ways.</i>	
<i>SFQL for structure.</i>	
<i>CD-RDx for intelligence.</i>	
<i>The next chapter.</i>	
RELEASE 0.5: HALF-BAKED NEWS	21
<i>ON's Instant Update.</i>	
PHONES/RESOURCES/CALENDAR	22-27
<i>Names, phones, dates.</i>	

there's so much more variety and complex structures to address: text objects such as footnotes, paragraphs, headlines; content-related items such as text categorization, indexing and search; creation and maintenance of links, cross-references and structures such as outlines/hierarchies, tables of contents and document identification. In addition, text may have display-oriented information: fonts and their sizes and styles; character sets; graphics, including vectorization of fonts and images; layout and formatting; hyphenation and justification. One system can rely on information provided by another; a document's representation depends on recognizing text objects, with headlines displayed one way and footnotes following a certain notation; a text-search program might search only the first three paragraphs of any document, or assign different weights to different parts of a document; a table of contents lists subheads.

All these are related at one level or another, but to handle them all at the same time would be foolish. The standards we're discussing here have to do only with text retrieval and content, not with display, layout, or other presentation and document-processing functions and issues addressed by standards such as Adobe's PostScript. In fact, the text-retrieval standards attempt to reduce the richness of text so that content can be specified according to a minimal syntax and texts retrieved by any client from any server.

Serve me some text

Basically, text can be retrieved in four ways -- by identity, by content, by association with other items (links, proximity, etc.), or by criteria.

Identity is very simple, or should be. A document is a specific piece of text, which can be assigned a unique ID number. But how can you keep all the servers from inadvertently reusing each other's IDs? Is John Quarterman's 1989 book *The Matrix* a version of his 1986 article "Notable Computer Networks" in *Communications of the ACM*? What about some of the chapters in it? Which is the *real* article about computers and privacy by John Markoff -- the one in the *New York Times*, or the slightly altered one that appeared later in the *San Jose Mercury*? The original or the translation? Do you want the 1989 projections, or the disappointing 1991 actuals for the same period?

Document IDs are important also for copyright records and other forms of authors' rights (cf. colorization, abstracts, and misquotations). They allow for authors to make specific references to other documents, including the server(s) where they may be found, and also could serve as the foundation for copyright protection and author-payment schemes. Ideally, IDs could save people repeating others' work since they could just incorporate it -- or annotate it, praise it, deride it or refute it -- by reference. You can also use a referenced document as the basis of a query without having to look at the document itself.

Content means "what it's about," and is the fuzziest but most universal description of a text; it's not unique or precise. Defining content perfectly is the unachievable ostensible goal of most text-retrieval systems. Content can be assessed by the presence of words, weighted by the presence of other words, etc. There are a variety of more complex ways of defining and assessing content (see **Release 1.0**, 3-90), including Verity's topic hierarchies, semantic analysis and thesauruses (semantic nets), and ranging all the way to natural language parsing, which may tell you what a text "says" as well as what it is talking about.

Associations is complex. It could be "all texts linked to 'bolt number 520J-Z2'." Or it could be "all articles cited in the footnotes in chapter 13" of a particular document. Or it could simply be items classified in a particular category, such as "life in the fast lane," rather than items containing those keywords.

Criteria are what would be called values in a database. These can include sources (publications, publishers, etc.), authors, dates of publication/copyright, and assigned, arbitrary classifications such as poetry or country of origin or editor's rating. In effect, criteria are associations with a category or value rather than with a specific object.

Obviously, these approaches slide into each other, and a search usually includes combinations of them. For example, you might want a section identified by content, within a book with a specific identity.

More broadly, there are two approaches -- unstructured text, where you're relying mostly on content, and structured text, where criteria and associations and defined elements are key. (Note that Juan's structure may be irrelevant or confusing or misleading to Alice; sometimes the goal of a search may be to find what nobody knew was there. Would Sherlock Holmes rely on information structure by Doctor Watson?) This distinction, although fuzzy, more or less corresponds to the difference between:

- on-line, dynamically changing information, where you usually search by content and there's likely to be a lot of redundancy (and large volumes of text to search: What's new in Leningrad? What are people saying about the new version of WidgeText? Let's find some articles that mention Graham Greene's years in Haiti.)
- CD-ROM, structured information, where you typically search by association or criteria for something in particular, perhaps a unique, specific answer: What happens if this bolt is unscrewed?¹ Let's see what our policy is on paternity leave for unmarried fathers.

However, text bases of periodicals and other random texts stored on CD-ROM (basically, on-line services on disk) tend to have the character of the first group. Of the three would-be standards discussed here, **WAIS (for Wide-Area Information Servers)** is oriented to on-line information, while **SFQL (Structured Full-text Query Language)** is oriented to structured CD-ROM information. The third, **CD-RDX (for CD Read-only Data eXchange)** is designed for CD-ROMs, but is better suited to unstructured information (or less optimized for structure) than SFQL. (Full details -- and qualifications of these generalizations -- begin on page 6.)

Text retrieval is more than just information for researchers and executives. It also supports tasks such as running help desks, deriving qualitative mea-

¹ The mechanic uses a hypertext text base to find out by reading what the engineer said. The engineer may use an object-oriented database with an engineering application to figure out the stresses and torques involved, and what other parts might get damaged or misaligned. And you may also need a database (OO or otherwise) to maintain the part's repair history.

sures for assessing press coverage, interpreting and responding to complaint letters, assembling precedents for legal cases or other decision-making processes, and many other "soft" tasks. Moreover, if you can specify a text object and procedures to act on text objects, you can automate a lot of work. Many publishing systems can automate previously direct-manipulation work not just in presentation and layout, but in conditional printing, document assembly, catalogue publishing and the like. But for now, we just want to be able to present them to a reader, who may then incorporate them into various text tools. Text-object definitions and the whole SGML/document-preparation world are a separate issue (despite derivative use of SGML by SFQL systems).

Client-server: The story so far...

The common notion of client-server is a database server, which supplies data -- generally data that can be specified and retrieved by SQL. Then you write client applications to do things to the data specified, and store the results back in the database, perhaps generating reports or invoices or bank statements along the way. Applications can also occur back at the server: stored procedures in a database, various kinds of other manipulations such as number crunching or image manipulation or polling of a physical measurement device.

Tools such as Agility's Wijit (Release 1.0, 11-90) or Sandpoint's Hoover, for access to public data services among other things, are designed to solve the text-retrieval (TR) interoperability problem. But they do so by building emulators/queries for each front-end to talk to each back-end. Agility/Dun & Bradstreet's John Landry notes the problems of continually changing back-ends, which vendors solve by updating their front-ends simultaneously. This creates few problems for their clients beyond updates, but big problems for companies such as Agility or third parties using and reselling the content. The standards discussed here would force the back-end vendors to hide their "innovations" behind an insulating layer that could interpret the standard protocol. (Wijit does the work at the client, creating the appropriate messages for each service it addresses and translating them back and forth into mail messages for the user; these TR standards would distribute the effort between client and server.)

But SQL is a productive aberration in the world of clients and servers. Most clients cannot talk to most servers. Instead, matched pairs communicate using proprietary protocols, getting the benefits of distributed data and access, optimized performance, and perhaps security or transaction management -- but not heterogeneous access. SQL was an important step to providing heterogeneous access: insulation of the specifics of one side from the specifics of another. Yet there are performance penalties and it's still rare for client and server to be developed and installed independently or to be moved around from server to server or client to client (although data does move). Most vendors and developers actually use supersets of SQL -- and thus are dependent on the features in the supersets.

Client-server applied to text

So how does text fit into this scheme? Text-oriented systems tools can benefit from the same sort of architecture, and from the same benefits of insulation through a common protocol, although the protocols themselves are different from SQL. Indeed, most text-search programs already use a rudimentary client-server architecture: The terminals are clients, and the hosts are servers. Most of the intelligence resides in the hosts, and requires a specific form of input from the clients, which are mostly dumbish terminals that know only how to log on and validate a request's syntax.

There are other kinds of examples, of course. For example, you can integrate a text client with a database to generate boilerplate letters. Or you can maintain a (relational) database of text objects, and use an expert system or a table as a client to assemble the components of a document. Saros Mezzanine is basically a SQL Server database of DOS files, each listed as a single record in the database, which can be found by attributes stored in the fields of each record. (The files themselves are stored outside the database, and incorporated only by reference.) Reach Networks uses a database to maintain a highly structured and linked set of text files.

And then there's Lotus Notes, which uses a tightly-coupled client-server architecture: The client knows the server data structures intimately, and vice versa. The benefit is that you can get specific pieces of text, arranged in specific ways such as outlines, tables, and chronological lists. You get the benefits of distributed access within a well-defined, homogeneous environment, but you lose the opportunity for access from heterogeneous systems. It's the usual trade-off between functionality and generality, as with applications written with SQL supersets. They use a common format for specifying the data, but the applications themselves are platform-dependent.

As noted, the goal is to have a protocol that can keep the front-end and the back-ends independent of each other. (We ignore the need for communications standards to establish contact in the first place. They are important and necessary, but not relevant to this discussion. It's assumed that you can establish a link, and that you have the proper authority and scripts to log on to any given service. Standards here would also be handy, but they are another issue.)

Three contenders

The three significant standards efforts in this area are immature and not widely known or effectively promoted. Each reflects the biases and needs of its originating community. You *may* be able to create a standard by committee, but you can get it adopted only through vigorous, effective marketing -- by people with vested interests who make more than token efforts to reach broad markets. Where are the 3Coms and Oracles² for these standards, to say nothing of the IBMs and Intels? Will Slate or someone else sell WAIS, SFQL and CD-RDx clients for PenPoint machines?

² Ethernet was a standard promulgated by Xerox, DEC and Intel, but 3Com was the independent start-up that proved its accessibility to everyone. SQL was created by IBM and adopted by ANSI, but it formed the basis of Oracle's business.

Many proponents of each standard are barely aware of the others. In part, this reflects the gulf between the on-line and the CD-ROM communities -- a gulf which itself reflects the immaturity of the whole field. Basically, the on-line people work with dynamic, continuously updated text and focus on content search (with some exceptions in the case of legal databases), and the CD-ROM people work with fixed, periodically updated texts with carefully architected structures and links. Thus it's appropriate that the content-oriented WAIS standard come from the library/on-line community and is based on its Z39.50 protocol for electronic card catalogues, while the structure-oriented SFQL approach comes from the CD-ROM/hypertext world of aircraft documentation. The third proposal, CD-RDx, also CD-ROM-oriented, is sponsored by the intelligence community for use on CD-ROMs with many varieties of data structures and types. (With the requisite plumbing, the CD-ROM protocols could of course be implemented for on-line access, and vice versa.)

Each group needs to expand outside its own community -- WAIS from the research/Internet community to commercial on-line services, SFQL from the aerospace industry to other commercial communities that could set industry data standards (insurance contracts? mortgages? construction plans?), and CD-RDx from government and a single vendor to commercial data suppliers.

COMPARE AND CONTRAST

	<u>WAIS</u>	<u>SFQL</u>	<u>CD-RDx</u>
<u>*Originating community</u>	libraries, info services	aerospace	gov't, intelligence community
<u>*Orig. medium</u>	on-line	CD-ROM	CD-ROM
<u>*Breadth</u>	wide	wide or narrow	wide or narrow
<u>*Model</u>	Z39.50	SQL	sui generis
<u>First implemented</u>	Z39.50 proto-types since 1986	2 interoperating c/s sets, Feb 90	Dept. of Commerce disk, 1990
<u>Current status</u>	WAIS NL systems at several sites	SQL2 demos later this year	version 3.1 shortly (DOS)
<u>Implementers</u>	one team w members from 4 companies	2 independent user companies	consulting firm
<u>Toolkits</u>	public domain source code	soon from Fulcrum, Scilab prototype	Helgerson, or do-it-yourself API spec
<u>*Structure</u>	none, optional	DTD/SGML, others	DTD, SGML, CALS, &c.

*The qualitative descriptions, marked by asterisks, indicate tendencies or most appropriate uses, but there are exceptions to everything. Both CD-RDx and SFQL will likely be used by NISO as the basis of an effort to develop a standard protocol for interface-independent retrieval. Z39.50 is a NISO standard, but the WAIS protocol differs significantly, much as SFQL differs from SQL. All can handle graphics and other non-text information.

The goal of all three is to allow any client to retrieve text from any server by using a simple protocol to specify texts by content, criteria or association, not by specific identity. The SFQL approach envisions a world of specific domains, where everyone is talking about, say, airplane parts; data structures and relationships are defined industrywide, but implemented differently on each server. The WAIS approach is more general and works across domains but without the power of SFQL; it could be used arbitrarily for searches across a wide range of Internet servers, news services, public or private databases, and possibly into SFQL servers with alternate front-ends. (An SFQL server would work in front of an unstructured text database, but it would be wasteful.) CD-RDx can handle either kind of data, using full-text search as necessary, but is implemented for use with CD-ROMs.

Thus these standards aren't so much competing as oriented to different but still overlapping tasks. One standard would be good, but insufficient; two or even three complementary standards would be much better. Twenty-nine (or is it 37?) "standards," the situation we have now, is a waste.

WAIS: MANY WAYS TO DO IT

WAIS is pronounced "ways" and stands for Wide-Area Information Servers. The "Wide-Area" aspect is secondary to (or easier to achieve than) the promise of heterogeneous access. WAIS is a project of four groups: Thinking Machines, the instigator, as a follow-on to its work with Dow Jones that created a text server for DowQuest (see **Release 1.0**, 1-88); Dow Jones News Retrieval, a content supplier; Apple Computer, focused on the interface; and KPMG, a highly involved user. The project leader is Brewster Kahle, a co-founder of Thinking Machines and also a virtual employee of Apple, where he spends a lot of time. The single greatest problem with this project as a standards effort is that it is being developed by a tight group of dedicated people; they tend to forget that they are trying to develop something wonderful rather than something general. However, there are now a lot of independent third parties using the WAIS source code to create WAIS servers and clients at some 150 universities, and 27 WAIS databases newly available over the Internet (too new to draw many conclusions from).

What is still missing is commercial commitments, but things look promising. Dow Jones is evaluating the WAIS pilot; KPMG found it extremely useful but doesn't have a wide-area network to use the service on a broad basis. Mead Data has participated in the implementation committee and is working on a WAIS prototype, but with no firm plans for it so far. "We need to have a published external interface for Mead's Nexis commercial news and information" (but not necessarily its structured Lexis legal service), says senior architect Peter Ryall. Other on-line vendors such as Dialog and CompuServe aren't active so far. Pandora Systems, a small consulting firm specializing in on-line access, plans to build a GeoWorks-based WAIS front-end, nicknamed the "cyberspace cockpit." His goal is to mimic the Apple interface (with permission) and extend it with facilities for managing access and filters for Internet news groups. Also, NeXT plans to incorporate WAIS as part of a broader information strategy which will include structured searches as well as the pure WAIS natural-language approach. NeXT is already using a prototype to work on access to a variety of sources, news feeds and relational databases, says NeXT's Adam Hertz.

The WAIS project itself is focused on providing idiot-proof, "natural-language" access to text, while the protocol standard is intended to support a variety of query methods, including Boolean or conceivably SFQL (below). The general part of the system is a small, simple protocol, based on a library-community ANSI-NISO (American National Standards Institute-National Information Standards Organization) standard called Z39.50-1988 (also proceeding within the International Standards Organization as DIS 10162 and DIS 10163, but nicknamed SR-1 for Search & Retrieval).

Type 1, the only subset of Z39.50 defined so far, is Boolean retrieval, typically applied against an electronic card catalogue, not against the full text itself. Active proponents of Z39.50, defined in 1988 but just now coming into use, include just about the entire US research library community -- the Library of Congress, the Online Computer Library Center (an early user of Tandem machines), the Research Libraries Group, Carnegie-Mellon, and the University of California.

Z39.50 gets a makeover

WAIS is a superset/subset of Z39.50 (originally defined as Type 3 but now probably going to be an extension of Type 1), with some subtle changes to broaden its reach and eliminate some of the powerful but restrictive features of the original. These extensions are likely to be adopted by the NISO committee and merged back into the Z39.50 standard. Clifford Lynch of the University of California's Division of Library Automation is a key person in the Z39.50 effort, and is also tracking the WAIS project closely as a leader in the NISO committee shepherding Z39.50's evolution.

Where Z39.50 was originally designed to search electronic catalogues, returning a list of titles and document IDs so that you could then select the ones you wanted from a list, the WAIS approach is more oriented to full-text and even multi-media. (For multi-media, the search routines look for text associated with the non-text items, which are retrieved separately by IDs.) Thus Z39.50's Boolean searches of defined fields in a card catalogue (or any other document) are still possible but are no longer an integral part of the spec, which passes through arbitrary strings for full-text search as a least common denominator.

Moreover, while the original Z39.50 server maintains the "state" of the session -- i.e., it knows what documents it has listed for the user and can then select those he picks from the list -- the WAIS spec requires the client to maintain that list. Then the client sends back the precise IDs of the documents he wants searched to select parts, or to retrieve in full.

The benefits are that a single server can handle a number of clients more effectively, since the server handles each client transaction by transaction, and that documents identified by unique ID in one transaction can be used in a query to another server as well as to the original one. The WAIS protocol also includes an optional procedure for relevance feedback, whereby you can send a document ID and optional subsetting parameters (paragraphs, range of bytes, etc.), which is transformed into a document by the system as the text of a query. Exactly how the document gets from server to server (and is paid for, if necessary) is an exercise left to the systems implementer, but logically it is possible.

Sending a message

The protocol transmits text strings to search for and specifies where. It can also handle instructions for which fields to search or Boolean constraints or relationships among words -- how close together they must be, ands and ors and nots, as well as criteria such as date of publication, author, publisher, type of publication, headlines or abstract, or within the full text. It supports Boolean constraints and criteria explicitly but optionally; it could also support almost any other format, including, in extremis, a phrase that said in effect, "now speaking SQL:" which would alert an SQL server at the other end to turn on its SQL parser. Other systems would simply interpret the words in the SQL query as words, and do their best to find relevant texts according to their own methods. In fact, you could even use WAIS for actions, such as ordering reprints, although not formal transactions (at least as far as WAIS is concerned).

The WAIS protocol allows any client and any server to communicate without crashing. Thus, in a natural-language query, there could be a lot of extraneous stuff: "I'm wondering how come OS/2 seems to get such a rotten deal in the press." Or, "I'd like to know about poems about Alice Haynes by Juan Tigar." On the other hand, a structured query could use defined fields unintelligible ("author," or "to" and "from") to the server that receives them. In practice, you're unlikely to query a news database by "addressee," as you might a mail server, but if you did, the news database would simply ignore the "to" field.

The protocol itself carries no high-level notions of relevance, concepts, categories or structure; the interpretation happens on either side (just as with SQL there's complex data structures on one side and complex application and display logic on the other). This, of course, is where WAIS is likely to meet its strongest objections -- from people who say, "Well, my front-end can do a lot more. Why should I dumb it down for this system?" In fact, WAIS can pass through intelligent, structured queries as well. Not even stop words are removed, so that you can have two interdependent systems communicating with each other unknown to the WAIS protocol. Matched clients and servers work better in concert, of course, but all can work together to some extent. The goal is for all these approaches to compete on a playing field leveled by WAIS.

How does WAIS compare with Xanadu, the information server designed by Ted Nelson and now owned by Autodesk? (See Release 1.0, 7-89). To the naked ear, they sound alike. But they aren't. Xanadu is a server; it maintains close control over the content, and is a way of publishing and assembling info and managing it at a more granular, ID-oriented level. With Xanadu, you specify or follow links to get the precise, unique thing. WAIS is a way of finding and distributing information that has already been published in a variety of formats. With WAIS, you describe, and get a number of possibilities. Of course, you could have a Xanadu-specific WAIS front-end to Xanadu, but if you addressed Xanadu with the WAIS default natural-language query you would lose Xanadu's full power.

The server responds

The server makes its best effort to answer the user's query and sends back a list of texts, identified fully according to the WAIS syntax, with an ID, a title, score, types and date. (The ID includes the originating source, the copyright owner, and a unique ID, as well as the server supplying the document and the ID given it by that server.) The user can then select from the list to receive the full content (or a specified subset) of the documents listed, or he can refine or modify the query (with relevance feedback or other constraints).

The documents are listed by title (either a specified title or the first line of text by default), in order of their scores. The scores measure relevance, according to algorithms that may vary from server to server. On a Boolean server, that might simply be the number of times a specific word appears in a document, or the number of times it appears divided by the number of words in the document, or it might be a 1 for "present"; on a Thinking Machines server, it might be a complex, proprietary ranking that involves weights, co-occurrences of words, etc. (see **Release 1.0**, 1-88 and 3-90). The type defines the document's format -- TEXT, PICT, TIFF, etc. -- an extensible list that could include spreadsheet files or voice annotations. WAIS has already extended Z39 to handle multimedia by handling larger files, parts of files, and "understanding" the vagaries of graphics and potentially sound or video formats. Obviously, the client needs the appropriate facilities to represent the objects retrieved to the user, but the protocol itself can handle anything digital.

Another defined type is WSRC (for Wais SouRGe), which includes IDs for documents located elsewhere and instructions for connecting to the other server(s) where they are located -- i.e., a sort of incorporation by reference. That means one server can act as an index/pointers for others -- or a yellow pages, if you will. WAIS also offers a standard way to describe servers. In terms of its contents, a server can describe itself in answer to a WAIS full-text query, but other information is useful too. For example, what protocols do you support? What networks are you on? Who owns you? Where are your documents from and how frequently are they updated? And of course, what are the charges? The description of servers is one good place to include pricing information, although some documents may be priced individually. (You might even be able to run a remote interface to American Information Exchange, **Release 1.0**, 7-90.)

How does the refinement of the query relate to the first version? In a Boolean system, it could be the addition of "and not Paris." In a more sophisticated one, "before 1985," referring either to dates within the text (although the system might also pick up "Section 1203" or "1625 feet") or the date of publication of the text to be retrieved. In another system, it might be, "more articles like the third one you selected, but nothing like the first on the list" [which concerns a different Alice Haynes]. In that case, the second query consists of all the words in the selected document.

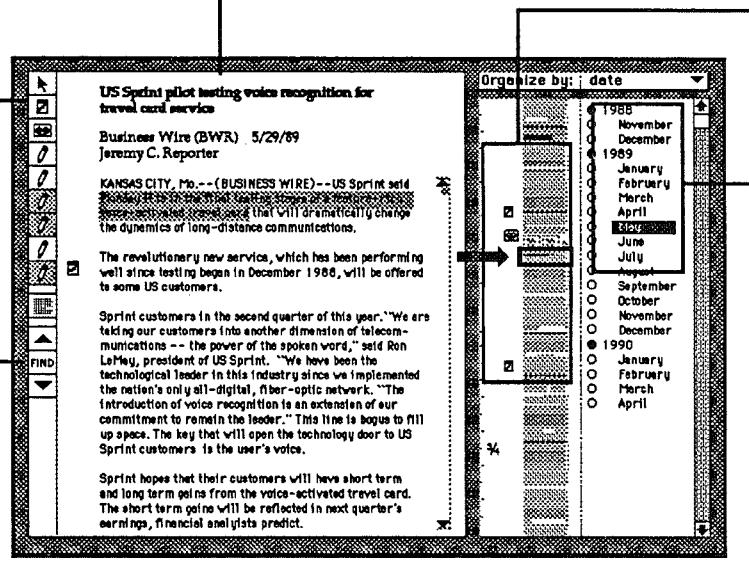
Behind the scenes at the server

The server may hold a variety of kinds of text bases, news groups, mail archives or bibliographies, and a variety of methods of finding things -- from a Connection Machine's brute-force string-searches to full-text indices to

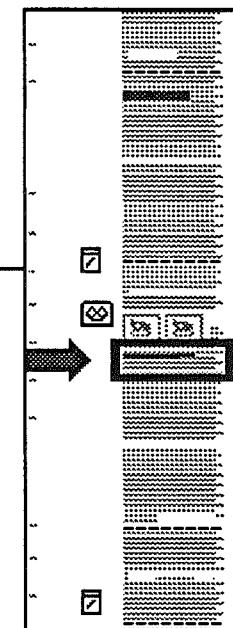
Annotation is supported through this palette of tools. The user is given access to (from top to bottom) "Posted" notes that can hold text data, a special type of Posted that can store audio annotations and a number of colored highlight pens.



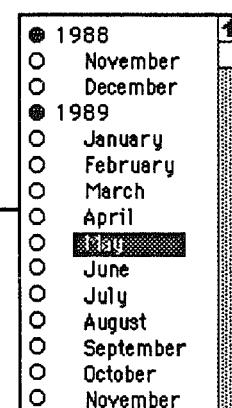
This central portion contains the "content" of the notebook – i.e. the actual data that was retrieved by the user.



The "Find" button and "next" and "previous" arrows allow the user to look for data based on a number of characteristics. The user can search for particular text strings. In addition, the user can select to search for earlier or later instances of particular highlight colors, "Posted" notes or audio annotations.



This 'bird's eye view' of the notebook allows the user to see a visual map of items in the vicinity of the current location. The large arrow marks the current location; the sizes of annotations are exaggerated. The user can quickly see that two images are immediately 'above' the current location, a highlighted passage is located farther 'above' and a "Posted" note is located 'below.' This view can also be used as a navigational device – by clicking on the desired location, the notebook content jumps to that location.



A hierarchical outline allows the user, in this case, to view the contents in chronological order. The user can expand the outline (e.g. 'open' a year into its months) or use it as a navigational device to jump to a particular section of the notebook. The user can also change the notebook's organization by selecting a new attribute from the "Organize by" menu at the top of the column.

Prototype design for information "notebook." This screen depicts a notebook in which a user can skim, search, organize and annotate information.

lists of articles and abstracts to a bulletin board of text items identified by keywords and classified into categories or news groups automatically or by a sysop, or selected as "editor's choices" by someone you revere. You could also have employee handbooks, automated help systems, on-line documentation, library catalogues, a database of patents with numbers and keywords and drawings, and so forth. The classification scheme could be anything from an alphabetical list of words (a plain index) to a hierarchy such as Verity's Topic, tailored for a certain subject, to a chronological file of mail messages to a highly structured text database such as Lotus Notes.

The WAIS project

The WAIS project comprises a number of separate interoperating installations, including a loaner Connection Machine at the KPMG New Jersey headquarters office that has now been returned to Thinking Machines. KPMG, the primary nontechnical user, experienced all the benefits other accounting firms have experienced with Notes and the Reach network (see **Release 1.0**, 2-91): better and more up-to-date information, better sharing of client contacts and corporate knowledge...overall a sort of automation and broadening of the old-boy network.

The user interface, "Rosebud," was developed by Apple's Advanced Technology Group, based on its earlier work on the interface on the Dow Jones DowQuest system. It allows users to type in natural language queries and to mark up the replies as yes, no, maybe, and select parts that are of particular interest. Those texts then constitute the basis of the second query (as supported by the protocol). Rosebud also includes some added features, as shown on the previous page. (This is from a paper Apple presented this week at the SIGCHI human interface meeting in New Orleans.) Another idea described is a "newspaper" which consists of a laid-out set of responses to a set of queries that are run daily: Thus each day you could get, for example, software news in the upper right-hand corner; John Sculley's daily activities in a box at the lower left; lacrosse on the left; and any mention of your own name featured in boldface type on top in the center.

The back-ends are Connection Machines, which perform high-speed parallel string searches and matching algorithms to retrieve the texts most relevant to each query. Other WAIS servers, such as those at universities, mostly use serial-search text engines and indexes. The WAIS server software will also shortly be installed on existing Connection Machines at Xerox PARC, at a shared site at Baylor and Rice Universities, and some other places. You can buy your own starter set for about \$150,000, software included.

Sharing the smarts

Like other client-server architectures, WAIS offers economies of scale. If you're doing something very smart, you can apply it on the server side, where anyone can use it through WAIS, rather than on the client side (where only a subset of customers will buy it). This assumes, of course, reasonable adoption of WAIS. The client-server separation allows the maximum intelligence in the model applied to the texts, and maximum access even from clients who don't know that model. Likewise, in general, it's best for the protocol to pass on the query in its full richness, rather than trying to interpret it. Clever clients can apply their cleverness across a multiplicity of servers.

The user interface helps in making the system intelligible to the user (rather than the user intelligible to the system, which is the server's job). On the server there's complex text, and possibly text-searching and categorization capabilities. On the client side, there's a complex human reader/editor/writer. But communication between the two sides is sparse. Thus the protocol provides the generality, and the systems on either two sides provide the richness and power.

Appendix: Still on the agenda

Issues of security and the like are up to each server/service. So are payments. Specifying costs is not yet part of the protocol, although this information can ride along through it. There are a number of possible pricing algorithms -- by time and time of day or week, by length or identity of items found or delivered, with charges potentially varying from document to document as well as server to server. Although many of the people spearheading this effort are of the free-information camp, it is vital for the spec to be broadened to include a way to specify charges. (They know this; they just forgot it when they get excited.)

Pricing information would make the protocol useful not just to libraries (which also need to cover their costs, rather than restrict access to other member libraries) but also to more commercial services such as those of Dow Jones, Reuters, Mead Data and hundreds of potential information suppliers who will be drawn into the broader market WAIS could foster. Rather than be a subscriber to a specific service, with an account name and a specific piece of front-end software acquired along with the subscription, one could be anyone with a valid credit card number -- and some positive identification, of course. The adoption of the WAIS standard, in fact, could be an important factor in the blossoming of the Electronic Frontier, with information traded freely (but not for free) among a wide community.

Free services can also be part of the same network. Indeed, we believe a properly competitive market will include both free and fee services. One early service, of course, will be a server of servers (Thinking Machines already offers one) -- an information service listing where you might want to search for certain kinds of information. Instead of texts, it will respond to queries with the names of likely servers for the information desired, in a format that the front-end can present to the user to select from for the search. (Pricing information will be included.) A smarter server, with pointers to the best articles on a particular topic -- basically, a selection editor as opposed to a copy editor -- could charge for its services. (See **Release 1.0**, 7-89, on hypertext publishing.)

There are also physical connection issues to resolve. Those can be handled by the client, which either will have the numbers of the servers desired, or know how to reach them over some internal or external mail network. Remember that WAIS is a spec; the implementation details will vary tremendously. It simply makes it possible for systems to interoperate, but the underpinnings have to be there. (Most of these issues also apply if the other two standards are used to communicate with on-line services.)

The sequel....

The consortium -- or rather, the informal project team behind WAIS -- hasn't yet begun any formal efforts to promote it. (Consider our coverage one of

the first such moves.) Accordingly, there's no groundswell of support yet. A few vendors are aware of the project, but most aren't au courant. Many consider it a proprietary effort on the part of Thinking Machines and Dow Jones. "They love the natural-language, relevance-feedback approach, of course," said one person we talked to, "because it takes a lot of machine power and Thinking Machines can do it better than anyone else." Although the protocol allows for intelligent searches, the hearts of this group are definitely with the naive user.

But all a standard needs is a broad front, not necessarily a consistent, united one. While the other two standards efforts described below are also significant, the role of WAIS as a means to communicate in almost real-time among people, rather than access to prepared, edited, structured data sources, makes it of more social, political importance than the other two.

SFXML: WHEN STRUCTURE COUNTS

The chief advantage of WAIS is its breadth and adaptability. It is also neutral; you can pass intelligent messages across it, but it's unaware of them. A different approach is that of SFML, which allows for independent clients and servers, by allowing them to communicate formally about the structure as well as the content of the data. (Or they may share a common, standard data schema specified by an outside authority, such as a trade group or anyone who controls both clients and servers.)

SFML is the product of a group of airline and aerospace companies and their vendors. It was driven by their need to publish, maintain and retrieve documentation for aircraft, which have components (most notably airframes and engines) from a variety of suppliers. One early effort was a customer's: British Airways, KnowledgeSet, Maxwell Data and Boeing got together to put documentation for BA's Boeing 757 aircraft onto CD-ROM in 1987. However, that system is closed; i.e., you can't use its software to retrieve any other vendor's documentation for any other Boeing aircraft -- or any other aircraft owned by BA.)

The BA project was one of the first; now this problem has become increasingly apparent. It's aggravated because engines and airframes come from different vendors, and some airlines contract maintenance out to other airlines. Typically, you need a separate system for each supplier, since each supplier builds its own CD-ROM documentation system in conjunction with one of several CD-ROM preparation houses. Moreover, BA has no wish to fund another such project; presumably, it would like its suppliers to provide documentation on CD-ROM in a format that could be read by front-ends from a variety of competing front-end system providers.

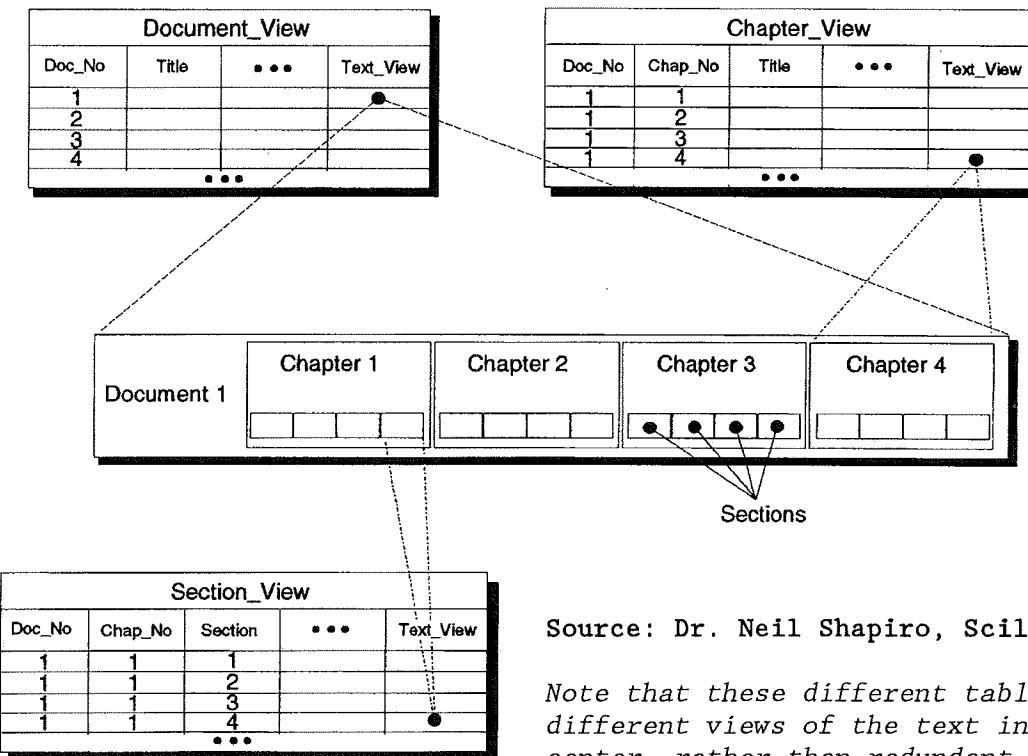
At the instigation of the Air Transport Association and the Aerospace Industries Association, a committee of customers and vendors for both equipment and software documentation systems got together to come up with a standard for interoperability -- and two separate, interoperable implementations. The group includes software vendors Context Corporation, EDS, Fulcrum, IBM, KnowledgeSet, Maxwell Data Management and TMS; ATA members American Airlines and British Airways; and AIA members Aerospatiale, Boeing, Douglas and GE.

What is SFQL?

SFQL stands for Structured Full-text Query Language, based on a subset of SQL (Structured Query Language). It leaves out relational database functions such as dynamic updates, joins, transaction management, dynamic view definitions and subqueries which don't (for now) seem relevant or cost-effective with text databases. The premise -- and power -- of SFQL is that the text being searched does have some structure, including such things as a title, an author, an abstract, headings and subheadings (which can be called out to produce a table of contents). There may also be cross-references between items, a topic index, versions and updates.

Full-text search is probably both too broad and too vague to handle these kinds of queries. Full-text search with relevance is quantitative, whereas with SFQL you can get precisely the right references -- rather than enough information to satisfy curiosity or a query. Compare the concrete relationship of a bolt to the fan it attaches to an engine, and the vaguer, discreet connection between Juan and Alice (they co-occur a lot, but their exact relationship is unknown -- and keeps changing). Moreover, SFQL can build (project, in relational terms) new text structures: You may want different subsets depending on whether your plane has two galleys or extra first-class seats.

Thus, SFQL implicitly turns the text into sets of tables, where each item is a record with a multiplicity of fields of arbitrary length (below). Just as



Source: Dr. Neil Shapiro, Scilab

Note that these different tables are different views of the text in the center, rather than redundant instances of it. The **_view** columns are simply different perspectives on the hierarchy created on the fly. The text tables could have been completely normalized into, say, a paragraph table, but would probably require excessive reconstruction of documents from little chunks. This is an implementation/optimization issue.

are simply different perspectives on the hierarchy created on the fly. The text tables could have been completely normalized into, say, a paragraph table, but would probably require excessive reconstruction of documents from little chunks. This is an implementation/optimization issue.

you can create a hierarchy from tables showing which items fall under which other items, so can you create a text database showing cross-references, components and so forth. Then you can use a superset of SFQL -- with the important concepts of "CONTAINS a string," subsets/sections of an entire document, and proximity of one term to another added -- to search it.

The initial version of SFQL dealt with the text as a simple concatenation of variable-length fields in a lengthy record; it supported both queries by criteria and full-text search within any or all fields ("contains..."). The newer version, SFQL2, now in final revision, can handle the more subtle (and appropriate to structured documents) notions of hierarchies and components and subcomponents -- although the schema is still maintained as tables, not as a logical hierarchy. That is, a paragraph is also part of a chapter; any text can contain a variety of separately specified fields such as part names or diagrams, cross-references can be maintained, and a listing of chapter headings can also be viewed as a table of contents. It all has to do with the ability of SQL (inherited by SFQL) to create views, so that the same item of text can be seen as itself, as part of a chapter, or as a collection of subsections. Headings can be collected into a view as a table of contents, and cross-references can be maintained as fields in yet other tables.

Vendors two

The original SFQL concept and spec were developed at GE's Corporate Research and Development Center by Neil Shapiro, now an independent consultant with his own firm, Scilab. Further work on it and SFQL2 was continued by Shapiro and Fulcrum of Ottawa and KnowledgeSet of Mountain View, CA. Fulcrum is uniquely suited to this task, since it's a long-time believer in client-server technology (its first full client-server toolset came out late last year after four years in development). The company isn't well-known outside the text-retrieval world because most of its software is sold through OEMs such as Siemens Nixdorf, HP, Data General, Sun, ICL, and NCR. Thus it has an API of almost 200 commands, a strong sense of openness, and the ability to build a server to implement the evolving specs of SFQL. Fulcrum gets about half its revenues from disk-oriented retrieval systems, and half its revenues from CD-ROM software; rather than consulting, it sells licenses to its engine to publishers or data-preparation houses. Fulcrum, with revenues of about \$5 million last year, is owned by Datamat, a systems house (and Fulcrum client) based in Rome.

KnowledgeSet brought to the party its intensive experience with British Airways and Boeing, along with KRS, an engine and flexible toolset for text preparation, and a complete user interface. (Fulcrum usually leaves the interface to its resellers, who integrate it with their own offerings.) KnowledgeSet is CD-ROM- and consulting-oriented; it specializes in building text-management systems to order. Somewhat smaller than Fulcrum, it is a subsidiary of Banta Corp. (which has revenues of \$660 million).

KnowledgeSet sees SFQL as a way into the aerospace market, but not one which it can afford to espouse without paid development contracts, its primary source of income. For Fulcrum, SFQL -- and openness in general since it sells a naked engine -- is more of a religion. The company plans to support SFQL in a forthcoming release of its software.

The two implementation teams, working separately, were Aerospatiale, using an engine from Fulcrum and GE, using the KRS engine from KnowledgeSet. Each

group developed both an information server with aircraft documentation and a separate Windows-based front-end. In fact, GE built two front-ends -- an interactive SFQL front-end where you would actually build a query in the SFQL syntax, and a forms-based front-end that dynamically loaded field names supplied at runtime by the server. Aerospatiale had a forms interface with field names based on the ATA 100 standard for documentation; it was easier to use but less flexible.

Ready, set, switch!

The great moment came last year at the February AIA/ATA meeting in Washington. Each team demonstrated its system. Then they switched disks, which contained both data and each team's server software (which also ran under DOS/Windows). They both still worked.

SGML, DTDs, schemas and OODBs

SGML, or Standard General Markup Language, is often described as an SQL for text. In fact, it's more like an SQL syntax and language generator; markup is only one example of the possibilities. That is, SGML is a small, extensible language that allows builder-users to build Document Type Definitions that describe the various allowable components of a specified document. The components within a document are "tagged," or identified as various elements in the DTD, so that they can later be manipulated by an application (for layout or display, for example) or by a database engine (for selective publishing or retrieval, for example).

Overall, a DTD is a framework for a document: There are DTDs for books, for documentation manuals, for government RFPs -- hence the government's interest, as expressed in the government's CALS (for Computer-aided Acquisition and Logistic Support) Initiative, for catalogues, and for a variety of other documents. The definitions can be strict or loose -- four sections with three subsections each, or a preface and several chapters followed by an index. There can also be content-specific tags, such as IDs for drug names or part names in documentation, or formats for identifying legal cases, or questions vs. answers. Figures can be identified and linked to text markers, and so forth.

The specific framework for defining and relating these components constitutes a DTD. Or they can be links to another text base, or even queries, so that a table could be automatically updated. Essentially, SGML is a tool for creating rich data/database definition languages, or DTDs. Beyond that, you can build a relational schema, such as the ATA 100 spec, using the elements of a particular DTD.

You could also store documents in an object-oriented database, which would maintain the intricate schema directly instead of representing it implicitly as sets of tables recording the structure. (In addition, an OODB manages the binding of methods to objects and other niceties.)

Strong or weak; tight or loose

The ATA Spec 100 standard includes a schema for aircraft documentation implemented in SFQL, but SFQL can actually be used more broadly. Just as an SQL database has a catalogue (which is a metadatabase about the database it manages), so does SFQL use a metadatabase, or schema, about the texts it manages. This schema can be part of a standard -- as in AIA/ATA -- or it can be built on a single server and downloaded to any front-end, thus providing enough information for any SFQL front-end to communicate intelligently with that SFQL back-end and its schema. Having a standard schema gives you the ability to create more tailored front-ends that make access easy for end-users (as Aerospatiale did), but the ability to define one dynamically gives you more flexibility overall, and means that SFQL ultimately can address a large range of information models and domains.

This text metadatabase is close to, or is a possible kind of, Document Type Definition, or DTD. DTDs are well-known in the text world. See box. The SFQL server converts the SGML document spec (which traditionally had to be parsed sequentially, from beginning to end, for the system to understand a document's structure) into a database structure. Then, you can search the document as a database rather than as an in-memory structure.

Thus there can be both tight or loose standards based on SFQL: the SFQL language itself, which is quite broad, and domain-specific SFQL/schema combinations such as the AIA/ATA standard. Given the issues with query optimization and the like, there will still be fierce competition among server providers, both for general performance and for efficient implementations of the data structures defined by specific DTDs.

CD-RDx: FROM THE ULTIMATE SPECIALISTS IN INFORMATION...

One of the biggest contributors to the development of text technology in the US has been the Central Intelligence Agency. It provided the initial funding for Xerox's hypertext tool, NoteCards, and was also a key customer for Verity's Topic. Now the Information Handling Committee of the Intelligence Community Staff, a sort of information-management coordinating body for the entire US intelligence community, is offering us CD-RDx.

CD-RDx is a spec designed at the request of the IHC by Helgerson Associates, a CD-ROM consulting firm headed by CD-ROM guru Linda Helgerson. An early implementation was fielded in the summer of 1990 on a disk of export-import information for the Commerce Department, and Helgerson is currently working on a second, improved implementation of a twice-improved spec (version 3.1), in response to feedback from government agencies and software vendors. A DOS server was delivered to the IHC this week, with a DOS client to follow in July. Versions for other operating environments are due later this year.

CD-RDx has its staunchest support from the intelligence community, DOD, and other government institutions such as NASA, GSA, Defense Mapping Agency, and the Patents and Trademark Office, which is desperately in need of a better way to classify and track patent filings (see **Release 1.0**, 8-89). The goal is to enable government units to share information easily, regardless of what vendors prepare the data or supply the software and the hardware.

The CD-RDx advisory panel are working on a spec with Helgerson Associates; Helgerson is working not just on an implementation, but on a number of versions of the server software to work on a variety of hardware platforms.

The resulting software is government-domain: That is, the implementations as well as the spec can be freely copied throughout the government and by its direct contractors. The hope, of course, is that the spec will also make its way out into the commercial world: Any vendor can use it, and sell its own toolkits and implementations of it (although Helgerson will have some advantages by virtue of being first). Since a lot of data is used by both government and commercial firms, this makes sense.

The CD-RDx vision of interoperability is broader than those of WAIS and SFQL -- the issue here is not just client-to-server interoperability, but also server-environment-to-indexed-data. That is, the goal is to build a range of compatible CD-RDx server engines so a variety of operating environments can all use the same sets of indexed data. In other words, an indexed data disk should be *platform-independent*. You can take a single disk and run it on a variety of hardware systems; the server software engine appropriate to the local operating environment will automatically load itself.

This is especially important for government agencies, which want to pass around indexed data from server to server among different agencies -- rather than commercial customers, who generally only want the same client to work with multiple servers, or on-line vendors, who want the same server to work for multiple customers. (On the other hand, CD-RDx vendors will find themselves able to address more platforms and thus more customers more easily.

Basically, CD-RDx is a set of APIs that can front-end almost any CD-ROM indexing scheme. It hides the specifics of an indexing system, but not the logical organization of the data or the fields and categories into which it's classified. Its APIs are akin to (but of course incompatible with) those of Fulcrum or a number of other vendors' -- commands to define and manage a variety of indexing schemes, download word lists, specify query terms and parameters, and so forth. Thus you can build a user interface that a user can use to query the server to see the kinds of data and search techniques he can use...and then he can use them.

Whereas SFQL implicitly supports a rich data schema (with all the overhead implied), CD-RDx is a little more pragmatic, and basically lets you talk directly to whatever indexing schemes and field structures happen to be around, without necessarily trying to integrate them into a single model. Matthew Goldworm of TerraLogics, a vendor of data preparation software with an orientation to maps, believes CD-RDx is more open to supporting maps and other data-rich structures than SFQL, which he considers too tied to the airline industry. In this aspect, CD-RDx has some of the flexible flavor of WAIS, but it also has more explicit support in the spec to address the specifics of any indexing scheme -- inverted text, table of contents, word and phrase lists, etc. That is, it is generally for building front-ends/applications to specific, structured data sets, rather than passing through ad hoc queries to a remote information service.

THE NEXT CHAPTER

So, how real is all this? We think it could be quite important if the right people get involved -- that is, commercial people with a vested interest in seeing it succeed, as well as the beneficiaries -- authors who will get wider, quicker distribution of their works, readers who will get broader but more precise access to the information they seek, and the world at large, because information will flow around with a little less friction. WAIS itself is simply a platform on which enterprising people will construct elaborate schemes for filtering, describing, pricing and distributing information. Profit, authors' pride and intellectual curiosity will provide the motivating forces, while WAIS is the machinery that will enable those forces to be harnessed.

We expect to see WAIS adopted from the library community out, with support from information providers pulled by users. WAIS will also benefit from the increasingly organized, broad community of information service users. As they get networked, they get more vocal, more organized, and better coordinated in making their voices heard. The electronic frontier is now being settled by people who have money and vested interests and the commercial force to make their voices heard.

On the other hand, in addition to the WAIS laissez-faire attitude, the world also needs standards for precise manipulation of structured information (which could in fact be transmitted via the WAIS protocol). Here, SFQL and CD-RDx are directly competitive. We expect SFQL to move from the aerospace community to other such industry groups, pulled mostly by intra-industry trade groups, with a push from software vendors such as Fulcrum. CD-RDx doesn't seem to have much momentum outside the government as yet, but those various government users may be able to get some commercial users and vendors excited.

Vendors tend to resist standards -- especially the leading vendors, who have commercial advantages and expect the world to adapt to them. Microsoft, for example, makes an analogy to SQL and likens its own CD-ROM standards to dBASE; it sees no need yet for a broader client-server standard such as SQL. Eventually, says Microsoft's Rob Glaser, SFQL will probably "work" for Microsoft, but right now he sees no need for it. This is an interesting analogy, given the recent impact of SQL on dBASE -- and questions about how history might have been different had Ashton-Tate been more open with dBASE (the Microsoft posture) or more open to SQL. The real question is: Will the standard of the future be Microsoft's, or will it be SFQL or CD-RDx or something else?

Overall, more and more users are beginning to use several information services and CD-ROMs and want a common interface. Rather than create a regulated industry (as with telephones) where you have one interface because you have one provider, we have the opportunity to create an industry of vigorous competitors operating with just one or two standard interfaces because that's what customers are asking for.

RELEASE 0.5 -- INSTANT UPDATE: THE USER KNOWS BEST

One of the advantages of the WAIS protocol discussed earlier in this issue is that it doesn't interfere with a user's best efforts to get what he wants. Although there's a lot of power in automation and groupware tools, people trying to work together frequently need facilitation rather than a fancy feature set. Working together should be made simpler, not "enhanced." Specifically, software shouldn't try to be any smarter than it can be. An excellent example of this principle is ON Technology's Instant Update.

Instant Update doesn't do much. It just lets people share virtual paper, update it, and pass it around. It flags conflicts but doesn't resolve them: The last one to update a paragraph (the basic unit within an Instant Update document) wins. It's not a fancy tool to edit share documents, nor a system to monitor people's movements, tell them what to do or manage conflicts.

But consider it in a more positive light: It's a way to send messages in context, like sticky paper for collecting feedback. Instead of getting answers to a question you've forgotten, you get updates to a shared memo. It may include a wild projection, a table of assignments, a calendar page, or anything that can be imported into a standard Mac document. It has the appeal of Post-It notes -- vanilla enough that they can do almost anything you can think of. When computers are truly ubiquitous, there's sure to be a copy of Instant Update on every refrigerator.

RESOURCES & PHONE NUMBERS

Kevin Tiene, Charles Bedard, **Apple Computer**, (408) 996-1010 or (408) 974-6433

Haviland Wright, **Avalanche Development**, (303) 449-5032

Holly DiMicco, **Boeing**, (206) 544-0990

Eben Kent, Barry Berkov, **CompuServe**, (614) 457-8600

Clare Hart, **Dow Jones**, (609) 520-5260

Paul Cotton, Peter Eddison, **Fulcrum**, (613) 238-1761, fax: (613) 238-7695

Linda Helgerson, Harvey Martens, **Helgerson Associates Inc.**, (703) 237-0682; fax, (703) 532-5447

Gary Ellis, **Information Access Company (Ziff-Davis)**, (415) 378-5278 or (800) 227-8431

Ed Rishko, **Information Handling Committee (Intelligence Community Staff)**, (202) 376-5560; fax, (202) 376-8003

Tom Rolander, **KnowledgeSet**, (408) 649-4193; fax, (408) 649-4692

Chris Bowman, **KnowledgeSet**, (415) 968-9888 or (800) 456-0469

Peter Ryall, **Mead Data Central**, (513) 865-7642

Rob Glaser, **Microsoft CD-ROM**, (206) 882-8080 or (206) 936-8294; fax, (206) 883-8101

Adam Hertz, **NeXT Inc.**, (415) 780-4579 or (415) 366-0900

Pat Harris, **National Information Standards Organization**, (301) 975-2814; fax, (301) 975-2128

Robin Palmer, **KPMG**, (408) 282-4272

Neil Shapiro, **Scilab**, (518) 393-1526; fax the same

Conall Ryan, **ON Technology**, (617) 876-0900

Michael Kinkead, **SandPoint**, (617) 868-4442

Neil Shapiro, **Scilab**, (518) 393-1526; fax the same

Matthew Goldworm, **TerraLogics**, (603) 889-1800

Brewster Kahle, **Thinking Machines**, (415) 329-9300 x228; fax, (415) 329-9329; brewster@think.com

Cliff Lynch, **University of California Library Automation Division**, (415) 987-0522/0526; lynch@postres.berkeley.edu

You can order a copy of the Z39.50 standard from NISO's distributor: Transaction Publishers, (908) 932-2280, for \$35.

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers PCs, software, CASE, groupware, text management, connectivity, artificial intelligence, intellectual property law. A companion publication, Rel-EAST, covers emerging technology markets in Central Europe and the Soviet Union. Editor & publisher: Esther Dyson; associate publisher: Daphne Kis; circulation & fulfillment manager: Lori Mariani; executive secretary: Denise DuBois; editorial & marketing consultant: William M. Kutik. Copyright 1991, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints or bulk purchases. Subscriptions cost \$495 per year, \$575 overseas.

RELEASE 1.0 CALENDAR

May 5-8 *Demo '91: The annual personal computer industry product review and demonstration - Palm Springs, CA. Sponsored by P.C. Letter. Call Kim Marker, (415) 592-8880.

May 6-7 Mobile Data conference - Cambridge. Sponsored by Waters Information Services. Call Betsy Martens, (607) 770-8535.

May 6-8 The 1991 Computer services & consultants executive conference - Orlando. Sponsored by IBM. With James Cannavino, George Conrades, Joseph Guglielmi, Ellen Hancock. Call Hal Topper, (404) 238-4228; overseas call Don Avery, 1 (416) 443-4606.

May 7-9 *National Online meeting - New York City. Sponsored by Learned Information. Call John Yersak, (609) 654-6266.

May 8 Massachusetts Computer Software Council's spring membership meeting - Boston. Keynote speaker: Steven Jobs. Call Joyce Plotkin, (617) 437-0600.

May 12-13 The thirteenth international conference on software engineering - Austin, TX. Sponsored by ACM, IEEE Computer Society. Call Barbara Smith, (512) 338-3336.

May 14-17 Quality Week 1991: Attaining realistic productivity and quality gains - San Francisco. Sponsored by Software Research. With Dr. Boris Beizer. Call Ed Miller, (415) 957-1441 or (800) 942-SOFT.

May 15 PC user group meeting - New York City. With Jerry Kaplan and Robert Carr, GO. Call John McMullen, (914) 245-2734.

May 19-22 *International Markup '91 - Lugano, Switzerland. Sponsor: Graphic Communications Association. SGML etc. Keynote by Esther Dyson. Call Joy Blake, (703) 519-8160.

May 19-22 IIA spring conference - Palm Springs, CA. Sponsor: Information Industry Ass'n. Call Linda Cunningham, (202) 639-8262.

May 19-23 International DB2 users group: Distributing the experience - San Francisco. Speakers include Chris Date, Michael Stonebraker. Call Larry Fleischman, (312) 644-6610.

May 20-23 Spring Comdex - Atlanta, GA. Sponsored by the Interface Group. Call Elizabeth Moody, (617) 449-6600. Includes Windows World; coincides with Interface/91.

May 21-23 UNIX & Open Systems: Applications, tools & solutions for the '90s - Santa Barbara. Sponsored by Patty Seybold, UniForum and X Open. With David Stone, DEC; Peter Weinberger, AT&T USL; Ira Goldstein, OSF; Pete Peterson, WordPerfect; Charles House, HP. Call Deborah Hay, (617) 742-5200.

May 21-23 Silicon Graphics developer's forum - San Francisco. Sponsored by Silicon Graphics. Call Debbie Chen, (415) 335-1392.

May 22-23 Investing in venture capital - New York City. Sponsored by the Institute for International Research. Call Tom Judge, (212) 826-1260 or (800) 345-8016.

May 27-31 Avignon '91: Expert systems & their applications - Avignon, France. Sponsored by AFIA, ARC, ECCAI & JSAI. Call Jean-Claude Rault, 33 (1) 4780-7000 or fax, 33 (1) 4780-6629.

May 28-30 Database World - Washington, DC. Co-sponsored by Digital Consulting and Government Computer News. Speakers include Charles Bachman, Robert Epstein, Umang Gupta, Jacob Stein. Call Tom Reiling, (508) 470-3880.

June 3-6 **Macworld Expo/Berlin** - Berlin, Germany. Sponsored by World Expo Corporation. Call Deborah Paul, (508) 879-6700.

June 3-7 ***Object World** - San Francisco. Co-sponsored by the Object Management Group and World Expo Corp. Businesspeople's answer to OOPSLA. Call Dave Bradway, (508) 820-8123.

June 4-5 **Customer care conference** - Chicago. Sponsored by Software Strategies. Speakers include Barbara Brizdle, Richard Brock, Pat Landry. Call John Jacobsen, (203) 335-6090.

June 4-6 ***Digital World** - Beverly Hills, CA. Sponsored by Seybold Seminars. Digital data meets media and communication industries. Speakers include Steven Jobs, Trip Hawkins, Robert Winter. Call Beth Sadler, (213) 457-5850.

June 9-12 ***2nd annual SPA European conference** - Cannes, France. Sponsored by SPA. Call Ken Wasch, (202) 452-1600.

June 9-12 **Expert Communications '91** - San Francisco. Sponsored by Graphic Communications Association and Davis Review. Call Mills Davis, (202) 667-6400 or Joy Blake, (703) 519-8160.

June 9-16 **Poznan international fair** - Poznan, Poland. US exhibits sponsored by Department of Commerce, Eastern Europe Business Information Center. Call Bill Vigneault, (202) 377-1793.

June 17-18 **Virtual Worlds: Real challenges** - Menlo Park, CA. Sponsored by SRI International, The David Sarnoff Research Center and VPL Research. Speakers include Jaron Lanier, VPL Research; Warren Robinett, University of North Carolina; John Thomas, NYNEX Corporation. Call Teresa Middleton, (415) 859-3382.

June 17-18 **Technical product development through strategic customer support** - San Francisco. Sponsored by the Institute for Int'l Research. Call Kathleen Erb, (212) 826-1260 or (800) 345-8016.

June 17-21 ***International Computer Forum** - Moscow. Sponsored by the International Computer Club. Call Levon Amdilyan, 7 (095) 921-09-02, or "levon" on MCI Mail at 439-1034; or Esther Dyson at 1 (212) 758-3434.

June 18-21 **Videotex 91: Broadening the consumer market** - Crystal City, VA. Sponsored by Videotex Industry Association. Call Debbie Tritle, (301) 495-4955.

June 19-21 **Supercomputing USA/Pacific '91** - Santa Clara. Sponsored by Meridian Pacific Group. Call Gerard Parker, (415) 381-2255.

June 24-27 **SCOOP East '91** - East Rutherford, NJ. Sponsored by the Wang Institute of Boston University and the Journal of Object Oriented Programming. Call Bob Daniels, (508) 649-9731.

June 24-28 **First international Windows 3.0 developers conference** - Santa Clara. Sponsored by The Wang Institute of Boston University. Keynote speakers include Bob Muglia, Microsoft; Eugene Wang, Borland International. Call Andree Fontaine, (508) 649-9731.

June 25-27 **PC EXPO** - New York City. Sponsor: Blenheim. With Ray Noorda. Call Annie Scully, (201) 569-8542 or (800) 444-EXPO.

June 25-27 **Multimedia '91** - London, UK. Sponsored by Blenheim Online. Call Lynne Davey, 44 (81) 868-4466.

July 2-4 ***Machine Translation Summit III** - Washington, DC. Sponsored by the Center for Machine Translation, Carnegie Mellon University. Call Jaime Carbonell, (412) 268-6591, e-mail: mtsummit@cs.cmu.edu.

July 9-14	*PC Forum - Moscow. Organized by IDG World Expo and Information Computer Enterprise, USSR; co-sponsored by several USSR state committees. Call Terence Coe, (508) 879-6700.
July 14-19	*AAAI conference - Anaheim. Sponsored by American Association for Artificial Intelligence. Also includes Innovative Applications of AI. Call Carol Hamilton, (415) 328-3123.
July 15-18	Network computing conference and exposition - Washington, DC. Sponsored by IDG World Expo Corporation. Call Brenda Cone, (800) 225-4698 or (508) 879-6700.
July 15-18	Communication Networks - San Francisco. Sponsored by World Expo. Keynotes: Mark Baker, British Telecom; Eric Schmidt, Sun Tech; Ambassador Bradley Holmes. Call Debra Anderson, (617) 769-8950 or (800) 225-4698.
July 17	*Software Entrepreneurs' Forum - Palo Alto, CA. Dinner talk by Esther Dyson. Call Barbara Cass, (415) 857-1110.
July 23-26	Artificial intelligence and the help desk - San Francisco. Sponsored by the Help Desk Institute. Call Elaine Worthington, (719) 531-5138.
July 28-Aug 2	*SIGGRAPH '91 - Las Vegas. Sponsored by ACM. Art meets computers: The place to see and be seen. Call Jackie Groszek, (312) 644-6610.
July 29-Aug 1	Tools U.S.A. '91 - Santa Barbara. Sponsored by Interactive Software Engineering. Call Bertrand Meyer, (805) 685-1006.
August 5-8	International workshop on human-computer interaction - Moscow. Sponsored by California State University and the International Centre for Scientific and Technical Information, Moscow. Contacts: Larry Press, (213) 475-6515, fax (213) 516-3664, e-mail lpress@venera.isi.edu; or Yuri Gornostaeve, 7 (095) 198-72-41 or enir@iaeal.bitnet.
August 6-9	Macworld Expo - Boston. Sponsored by World Expo Corporation. Call Deborah Paul, (508) 879-6700.
August 11-13	*GeoCon/91 - Cambridge, MA. Sponsored by Soft-letter. An international product showcase for European, Canadian, Asian and Latin American developers who seek U.S. publishing or partnership contacts. Call Jeff Tarter, (617) 924-3944.
August 14-16	Windows & OS/2 - Boston. Co-sponsored by PC Week and CM Ventures. Call John Bourgein, (415) 601-5000.
September 4-6	UNIX Open Solutions - San Jose. Sponsor: Interface Group. Call Elizabeth Meagher, (617) 449-6600 or (800) 325-8850.
September 11-13	Breakaway 1991 - Atlantic City, NJ. Sponsored by ABCD. Resellers and vendors trade tips and "frank discussion." Call Debbie Keating, (601) 977-9033.
September 11-14	Software Publishers Association annual conference - Orlando. Sponsored by SPA. Call Ken Wasch, (202) 452-1600.
September 12-14	*ETRE - Opio, France. Sponsored by Dasar. Call Alex Vieux, (415) 321-5544.
September 20-21	Sources 1991: Asian financing & alliances - Santa Clara. Sponsored by Asian American Manufacturers Association. Call George Koo, (415) 321-AAMA.
September 22-24	*Agenda 92 - Laguna Niguel, CA. Sponsored by P.C. Letter/PCW Communications. Call Tracy Beiers, (415) 592-8880.
September 25-27	*Second European conference on computer-supported cooperative work - Amsterdam. Knowledge workers and academics, unite! Organized by the Center for Innovation and Cooperative Technology of the University of Amsterdam. (The language of

cooperation is English.) Call Mike Robinson or Liam Bannon, 31 (20) 525 1250/1225; fax, 31 (20) 5251211; e-mail, Bannon@learn.ucd.ie; or Charlie Grantham, 1 (415) 370-174; cegrant@well.sf.ca.us.

Sept 30-Oct 1 **Virtual Reality conference** - San Francisco. Sponsored by the Meckler Corporation. Call Marilyn Reed, (203) 226-6967 or (800) 635-5537.

Sept 30-Oct 4 ***Seybold Conference** - San Jose. The leading event in the computer publishing community. Sponsored by Seybold Seminars/Ziff. Call Kevin Howard or Beth Sadler, (213) 457-5850.

October 1-3 **INFO '91** - New York City. Sponsored by Cahners Exposition Group. Call Marilyn Harrington, (203) 352-8477.

October 1-4 **Seybold computer publishing conference & exposition** - San Jose. Sponsored by Seybold Seminars. The evolving process of communication. Call Beth Sadler, (213) 457-5850.

October 6-11 ***OOPSLA '91** - Phoenix. Sponsored by ACM. Call John Richards, (914) 784-7731.

October 7-11 **Interop '91** - San Jose. Sponsored by Advanced Computing Environments/Ziff. With Ellen Hancock, IBM Communication Systems. Call Dan Lynch, (415) 941-3399.

October 14-18 **CD-ROM Expo** - Washington, DC. Sponsored by World Expo Corporation. Call Terry Merrell, (508) 879-6700.

October 15-17 **NetWorld '91** - Dallas. Sponsored by Bruno Blenheim. Call Annie Scully, (201) 569-8542 or (800) 444-EXPO.

October 17-21 **USA Showcase '91** - Budapest. Co-sponsored by the Hungarian Ministry of Trade, the Hungarian Chamber of Commerce and the American Chamber of Commerce in Budapest. Call Jay Bowman at (713) 266-0610.

October 21-23 **Twelfth annual Alex. Brown technology seminar** - Baltimore. Primarily for investors. Call Lori Bresnick, (301) 727-1700.

October 21-25 ***Comdex** - Las Vegas. So wonderful they couldn't wait until November? Whatever the reason.... Sponsored by the Interface Group. Call Elizabeth Moody, (617) 449-6600.

October 27-29 **The Classic** - Monterey, CA. Sponsored by the American Electronics Association, for cute companies and eager investors. Call Flo Lewis, (408) 987-4200.

Oct 30-Nov 1 **UNIX Expo** - New York City. Sponsor: Blenheim Expositions. Keynote by Steve Jobs. Call Pam O'Neill, (512) 343-1111.

November 4-7 **ADAPSO fall management conference** - San Francisco. Sponsored by ADAPSO. Call Shirley Price, (703) 284-5355.

November 10-13 ****Second East-West High-Tech Forum** - Warsaw (Prague in 1992). Sponsored by EDventure Holdings. With a roster of serious-minded entrepreneurs and vendors from East and West. Don't just come to listen to advice; come to mingle with the people making it happen. Call Daphne Kis, 1 (212) 758-3434 or fax (212) 832-1720; MCI Mail: EDventure, 443-1400.

November 12-14 **Unicom '91** - Washington, DC. Sponsored by North American Telecommunications Ass'n. Call Susan Ryba, (202) 296-9800.

November 13-15 ***ComExpo Hungary '91** - Budapest. Sponsored by the Hungarian Telecommunications Scientific Society. Call Karen Venticiglia, (703) 527-8000.

November 17-20 **IIA annual convention** - Orlando. Sponsor: Information Industry Ass'n. Call Linda Cunningham, (202) 639-8262.

November 19-21 **PC Expo** - Chicago. Sponsored by Bruno Blenheim. Call Steve Feher, (201) 569-8542 or (800) 444-EXPO.

December 2-4 ***Alliance '91** - Tokyo, Japan. Sponsored by Harvard Business School Ass'n. Strategic alliances with Japanese companies. Call Mark Francis or Yasuhito Mikamo, (415) 742-0757.

December 3-5 **European Publishing conference** - The Hague, Holland. Sponsored by Seybold Seminars. Contact: Laurel Brunner, 44 (323) 410561 or fax, 44 (323) 410279.

December 15-18 ***Hypertext '91** - San Antonio, TX. Third international conference on hypertext. Sponsored by ACM. Call Janet Walker, (409) 845-0298, e-mail leggett@bush.tamu.edu.

Please let us know about any other events we should include. -- Denise DuBois

**The asterisks indicate events we plan to attend. Lack of an asterisk is no indication of lack of merit.*

RELEASE 1.0

S U B S C R I P T I O N F O R M

Please enter my subscription to **Release 1.0** at the rate of \$495 per year in the U.S. and Canada. Overseas subscriptions are \$575, airmail postage included. Payment must be enclosed. Multiple-copy rate on request. Satisfaction guaranteed or your money back.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

How did you hear about **Release 1.0**? _____

Payment enclosed.

American Express # _____ - _____ - _____ Expires _____

Signature _____

Please send me additional information on your multiple-copy rate.

Please fill in the information above and send to:

EDVENTURE HOLDINGS INC.
375 PARK AVENUE, SUITE 2503
NEW YORK, NY 10152

491

If you have any questions, please call us at (212) 758-3434.

Daphne Kis
Associate Publisher